

Isomorphisms of scattered automatic linear orders

Dietrich Kuske

Institut für Theoretische Informatik, Technische Universität Ilmenau, Germany

Abstract

We prove the undecidability of the existence of an isomorphism between scattered tree-automatic linear orders as well as the existence of automorphisms of scattered word automatic linear orders. For the existence of automatic automorphisms of word automatic linear orders, we determine the exact level of undecidability in the arithmetical hierarchy.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Automatic structures, isomorphism, automorphism

Digital Object Identifier 10.4230/LIPIcs.CSL.2012.455

1 Introduction

Automatic structures form a class of computable structures for which a number of interesting problems is decidable: while, due to Rice’s theorem, nothing is decidable about a computable structure (given as a tuple of Turing machines), validity of first-order sentences is decidable in automatic structures (given as a tuple of finite automata). This property of automatic structures was first observed and exploited in concrete settings by Büchi, by Elgot [12], and by Epstein et al. [13]. Hodgson [16] attempted a uniform treatment, but the systematic study really started with the work by Khoussainov and Nerode [19] and by Blumensath and Grädel [3, 4]. Over the last decade, a fair amount of results have been obtained, see e.g. the surveys [29, 1] as well as the list of open questions [20], for very recent results not covered by the mentioned articles, see e.g. [5, 11, 18, 17].

A rather basic question about two automatic structures is whether they are isomorphic. For ordinals and Boolean algebras, this problem was shown to be decidable together with a characterisation of the word-automatic members of these classes of structures. On the other hand, already Blumensath and Grädel [4] observed that this problem is undecidable in general. In [21], it is shown that the isomorphism problem is Σ_1^1 -complete; a direct interpretation yields the same result for successor trees, for undirected graphs, for commutative monoids, for partial orders (of height 2), and for lattices (of height 4) [27]. Rubin [28] shows that the isomorphism problem for locally finite graphs is complete for Π_3^0 . In [24], we show in particular that also the isomorphism problems of order trees and of linear orders are Σ_1^1 -complete. For the handling of linear orders, our arguments rely heavily on “shuffle sums”. Consequently, we construct linear orders that contain a copy of the rational line (a linear order not containing the rational line is called scattered, i.e., our result is shown for non-scattered linear orders). This is unavoidable since we also show that the isomorphism problem for scattered and word-automatic linear orders is reducible to true arithmetic (i.e., the first-order theory of $(\mathbb{N}, +, \cdot)$) and therefore much “simpler” than the isomorphism problem for arbitrary linear orders. But it is still conceivable that the isomorphism problem for scattered linear orders is decidable.

In this paper, we deal with automatic scattered linear orders. In particular, we prove the following three results:



© Dietrich Kuske;
licensed under Creative Commons License NC-ND
Computer Science Logic 2012 (CSL’12).

Editors: Patrick Cégielski, Arnaud Durand; pp. 455–469



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- (1) There is a scattered linear order whose set of tree-automatic presentations is Π_1^0 -hard (i.e. one can reduce the complement of the halting problem to this problem). This holds even if we fix the order relation on the set of all trees (Theorem 13). Hence also the isomorphism problem for tree-automatic scattered linear orders is Π_1^0 -hard (Corollary 14).
- (2) The existence of a non-trivial automorphism of a word-automatic scattered linear order is Σ_1^0 -hard (i.e. the halting problem reduces to this problem, Corollary 6). Again, this holds even if we fix the linear order on the set of all words (Theorem 5). The existence of an automatic non-trivial automorphism is Σ_1^0 -complete.
For regular languages ordered lexicographically, the existence of a non-trivial automorphism is decidable (Corollary 2), but it becomes undecidable for deterministic context-free languages (Theorem 8).
- (3) The existence of a non-trivial automorphism of a tree-automatic scattered linear order is Σ_2^0 -hard (i.e., one can reduce the set of Turing machines that accept a finite language to this problem, Theorem 17).

The proof of (2) uses an encoding of polynomials similarly to [24] but avoids the use of shuffle sums. The technique for proving (1) and (3) is genuinely new: One can understand a weighted automaton over the semiring $(\mathbb{N} \cup \{-\infty\}; \max, +)$ as a classical automaton with a partition of the set of transitions into two sets T_0 and T_1 . The behavior of such a weighted automaton assigns numbers to words w , namely the maximal number of transitions from T_1 in an accepting run on the word w . Krob [23] showed that the equivalence problem for such weighted automata is Π_1^0 -complete. The hardness results from (1) are based on a sharpening of Krob's result that can be found at [10]: there is a fixed weighted automaton such that the set of equivalent weighted automata is Π_1^0 -hard (and therefore undecidable). A closer analysis of this proof, together with the techniques for proving (1) and (2), finally yields (3).

These results show that the existence of isomorphisms and of automorphisms is nontrivial for scattered linear orders that are described by word and tree automata, resp.

A complete version of this extended abstract can be found as arXiv:1204.5653.

2 Preliminaries

2.1 Tree and word automatic structures

Let Σ be some alphabet. A Σ -tree or just a *tree* is a finite partial mapping $t: \{0, 1\}^* \rightarrow \Sigma$ such that $u0 \in \text{dom}(t)$ implies $u \in \text{dom}(t)$, and $u1 \in \text{dom}(t)$ implies $u0 \in \text{dom}(t)$ (note that we allow the empty tree \emptyset with $\text{dom}(\emptyset) = \emptyset$). A *(bottom up) tree automaton* is a tuple $\mathcal{A} = (Q, \iota, \Delta, F)$ where Q is a finite set of *states*, ι is the initial state, $\Delta \subseteq Q \times \Sigma \times Q^2$ is the *transition relation*, and $F \subseteq Q$ is the set of *final states*. A *run* of the tree automaton \mathcal{A} on the tree t is a mapping $\rho: \text{dom}(t) \rightarrow Q$ such that

$$(\rho(u), t(u), \rho'(u0), \rho'(u1)) \in \Delta \text{ with } \rho'(v) = \begin{cases} \rho(v) & \text{for } v \in \text{dom}(t) \\ \iota & \text{otherwise} \end{cases}$$

holds for all $u \in \text{dom}(t)$. The run ρ is *accepting* if $\rho'(\varepsilon) \in F$. The *language of the tree automaton* \mathcal{A} is the set $L(\mathcal{A})$ of all trees t that admit an accepting run of \mathcal{A} on t . A set L of trees is *regular* if there exists a tree automaton \mathcal{A} with $L(\mathcal{A}) = L$.

It is convenient to understand a *word* as a tree t with $\text{dom}(t) \subseteq 0^*$ (then $t(\varepsilon)$ is the first letter of the word). Nevertheless, we will use standard notation for words like uv for the concatenation or ε for the empty word. A *word automaton* is a tree automaton

$\mathcal{A} = (Q, \iota, \Delta, F)$ with

$$(q, a, p_0, p_1) \in \Delta \implies p_1 = \iota \text{ and } q \neq \iota.$$

This condition ensures that word automata accept words, only.

Let t_1, \dots, t_n be trees and let $\# \notin \Sigma$. Then $\Sigma_\# = \Sigma \cup \{\#\}$ and the *convolution* $\otimes(t_1, t_2, \dots, t_n)$ or $t_1 \otimes t_2 \otimes \dots \otimes t_n$ is the $\Sigma_\#^n$ -tree t with $\text{dom}(t) = \bigcup_{1 \leq i \leq n} \text{dom}(t_i)$ and

$$t(u) = (t'_1(u), t'_2(u), \dots, t'_n(u)) \text{ with } t'_i(u) = \begin{cases} t_i(u) & \text{if } u \in \text{dom}(t_i) \\ \# & \text{otherwise.} \end{cases}$$

Note that the convolution of a tuple of words is a word, again. For an n -ary relation R on the set of all trees, we write R^\otimes for the set of convolutions $\otimes(t_1, \dots, t_n)$ with $(t_1, \dots, t_n) \in R$. A relation R on the set of all trees is *automatic* if R^\otimes is a regular tree language.

Let $\mathcal{S} = (L; R_1, \dots, R_k)$ be a relational structure such that L is a set of trees. Then \mathcal{S} is *tree-automatic* if the tree languages L and R_i^\otimes for $1 \leq i \leq k$ are regular. The structure \mathcal{S} is *word-automatic* if, in addition, L is a word language. A tuple of tree automata accepting L and R_i^\otimes for $1 \leq i \leq k$ is called a *tree- or word-automatic presentation* of the structure \mathcal{S} .

2.2 Linear orders

For words u and v , we write $u \leq_{\text{pref}} v$ if u is a prefix of v . Let Σ be some set linearly ordered by \leq . Then \leq_{lex} denotes the lexicographic order on the set of words Σ^* : $u \leq_{\text{lex}} v$ if $u \leq_{\text{pref}} v$ or there are $x \in \Sigma^*$, $a, b \in \Sigma$ with $xa \leq_{\text{pref}} u$, $xb \leq_{\text{pref}} v$, and $a < b$. From the lexicographic order on Σ^* , we derive a linear order (denoted \leq_{lex}^2) on the set $\Sigma^* \otimes \Sigma^*$ of convolutions of words by

$$u \otimes v \leq_{\text{lex}}^2 u' \otimes v' :\iff u <_{\text{lex}} u' \text{ or } u = u', v \leq_{\text{lex}} v'.$$

By \leq_{lex} , we denote the *length-lexicographic order* defined by $u \leq_{\text{lex}} v$ if $|u| < |v|$ or $|u| = |v|$ and $u \leq_{\text{lex}} v$. We extend this linear order \leq_{lex} to trees. Let t be a tree. Then $t|_{0^*}$ (more precisely, $t|_{(0^* \cap \text{dom}(t))}$) is a word that can be understood as the “main branch” of the tree t . For $u \in \{0, 1\}^*$, let $t|_u$ denote the subtree of t rooted at u (i.e., $\text{dom}(t|_u) = \{v \mid uv \in \text{dom}(t)\}$) and $t|_u(v) = t(uv)$ for $u \in \{0, 1\}^*$ as well as $t|_u = \emptyset$ for $u \notin \text{dom}(t)$. Furthermore, $\tau(t)$ is the tuple of “side trees” of t , namely

$$\tau(t) = (t|_{0^i 1})_{0^i \in \text{dom}(t)}.$$

We now define the extension \leq_{trees} of \leq_{lex} to trees setting $s <_{\text{trees}} t$ if and only if

- $s|_{0^*} <_{\text{lex}} t|_{0^*}$ or
- $s|_{0^*} = t|_{0^*}$ and there exists i (with $0^i \in \text{dom}(s)$) such that $s|_{0^j 1} = t|_{0^j 1}$ for all $0 \leq j < i$ and $s|_{0^i 1} <_{\text{trees}} t|_{0^i 1}$.

In other words, we first compare the main branches of the trees s and t length-lexicographically and, if they are equal, compare the tuples $\tau(s)$ and $\tau(t)$ (length-)lexicographically (based on the extension \leq_{trees} of the length-lexicographic order to trees). Since the “side trees” $t|_{0^j 1}$ of any tree t are properly smaller than the tree itself, the relation \leq_{trees} is well-defined. Note that all the order relations \leq_{pref} , \leq_{lex} , \leq_{lex}^2 , \leq_{lex} , and \leq_{trees} are automatic.

A linear order \mathcal{L} is *scattered* if there is no embedding of the rational line $(\mathbb{Q}; \leq)$ into \mathcal{L} . Examples of scattered linear orders are the linear order of the non-negative integers ω , of

the non-positive integers ω^* , or the linear order of size $n \in \mathbb{N}$ that we denote \underline{n} . If Σ is an alphabet with at least 2 letters, then $(\Sigma^*; \leq_{\text{lex}}) \cong \omega$ is scattered, too. On the other hand, if $a, b \in \Sigma$ are distinct letters, then $(\{aa, bb\}^*ab; \leq_{\text{lex}})$ is countably infinite, dense, and without endpoints, i.e., it is isomorphic to $(\mathbb{Q}; \leq)$ [6]. Hence $(\Sigma^*; \leq_{\text{lex}})$ is not scattered. From [22, Prop. 4.10], we know that the set of word-automatic presentations of scattered linear orders is decidable.

A linear order $\mathcal{L} = (L; \leq)$ is *rigid* if it does not admit any non-trivial automorphism, i.e., if the identity mapping $f: L \rightarrow L : x \mapsto x$ is the only automorphism of \mathcal{L} . The linear orders ω , ω^* , and \underline{n} for $n \in \mathbb{N}$ are all rigid. On the other hand, $(\mathbb{Q}; \leq)$ as well as $(\mathbb{Z}; \leq)$ are not rigid.

Note that automorphisms of tree-automatic linear orders are binary relations on the set of all trees. Hence it makes sense to speak of an *automatic automorphism*. A tree-automatic structure is *automatically rigid* if it does not have any non-trivial automatic automorphisms.

Let $\mathcal{I} = (I; \leq)$ be a linear order and, for $i \in I$, let $\mathcal{L}_i = (L_i; \leq_i)$ be a linear order. Then the \mathcal{I} -sum¹ of these linear orders is defined by

$$\sum_{i \in (I; \leq)} \mathcal{L}_i = \left(\bigsqcup_{i \in I} L_i; \bigcup_{i \in I} \leq_i \cup \bigcup_{\substack{i, j \in I \\ i < j}} (L_i \times L_j) \right).$$

Intuitively, this \mathcal{I} -sum is obtained from the linear order \mathcal{I} by replacing every element $i \in I$ by the linear order $(L_i; \leq_i)$.

For $\sum_{i \in \mathbb{Z}} \mathcal{L}_i$, we simply write $\mathcal{L}_1 + \mathcal{L}_2$. If, for all $i \in I$, $\mathcal{L}_i = \mathcal{L}$, then we write $\mathcal{L} \cdot \mathcal{I}$ for $\sum_{i \in (I; \leq)} \mathcal{L}_i$. As an example, consider the linear order $\delta = \omega \cdot \omega^*$. This linear order is a descending chain of ascending chains. It will be used as “delimiter” in our constructions. Note that

$$\delta \cong (10^+1^+0; \leq_{\text{lex}})$$

where we assume $0 < 1$. Also for later use, we next define a regular set $D = \{t_{i,j} \mid i, j \geq 0\}$ of trees such that $\delta \cong (D; \leq_{\text{trees}})$. The alphabet of these trees will be the singleton $\{\$$ so that a tree is completely given by its domain. Then set inductively

$$\text{dom}(t_{0,j}) = \{\varepsilon, 0, 00\} \cup 1\{0^k \mid 0 \leq k \leq j\} \text{ and } \text{dom}(t_{i+1,j}) = \{\varepsilon, 0, 00\} \cup 01 \text{dom}(t_{i,j})$$

The trees $t_{0,4}$ and $t_{2,2}$ are depicted in Figure 1 (left-arrows denote 0-sons, right-arrows denote 1-sons).

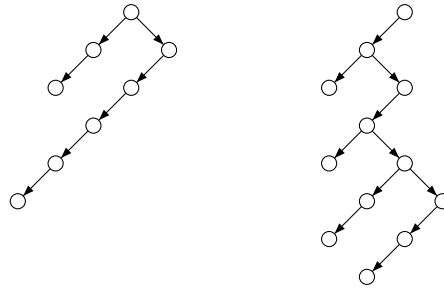
3 Automorphisms of linear orders on word languages

In this section, we consider linear orders on sets of words. The universe will be regular or context-free and the order will mainly be the lexicographic order \leq_{lex} and its relative \leq_{lex}^2 .

3.1 Regular universe and \leq_{lex}

In this section, we will show that the rigidity of a linear order $(L; \leq_{\text{lex}})$ with L regular is decidable. Even more, we show this decidability for regular words, i.e., word-automatic

¹ Shuffle sums mentioned in the introduction are special cases of this construction where $\mathcal{I} = (\mathbb{Q}; \leq)$ is the rational line and, for every $q \in \mathbb{Q}$, the set $\{r \in \mathbb{Q} \mid \mathcal{L}_q \cong \mathcal{L}_r\}$ is dense.



■ **Figure 1** Two trees from D .

structures of the form $(L; \leq_{\text{lex}}, (P_a)_{a \in A})$ with L and $P_a \subseteq L$ regular languages of words for all $a \in A$. The study of these regular words was initiated by Courcelle [8] who was interested in the frontier of regular trees. Thomas proved that their isomorphism problem is decidable [30] (an alternative proof was given by Bloom and Ésik [2]) and the complexity of this problem was determined by Lohrey and Mathissen [25].

The outline of our proof is as follows (missing definitions are given below): Let ν be some regular word given by a tuple of word automata. The basic observation is that ν is rigid if and only if all its \sim -equivalence classes as well as the quotient ν/\sim are rigid. This allows to do induction since, after finitely many divisions of ν by \sim , we end up with a single \sim -equivalence class. The central problem therefore is to determine the \sim -equivalence classes (up to isomorphism, there are only finitely many) as well as the quotient ν/\sim and to decide whether a single \sim -equivalence class is rigid. For these calculations, we first represent the regular word ν by a “term” (Heilbrunner [15]) and then transform such terms using a technique by Bloom and Ésik [2].

An *extended word* is a labeled linear order with a finite set of labels. A *regular word* over the alphabet A is an extended word $(L; \leq, \lambda)$ with $\lambda: L \rightarrow A$ such that

- L and $\lambda^{-1}(a)$ for $a \in A$ are regular word languages over some alphabet Σ and
- \leq is the lexicographic linear order \leq_{lex} .

A *term* over A uses constants $a \in A$ (standing for the extended word on $\underline{1}$ whose only element is labeled a) and the following operations:

- concatenation of words (denoted $\mu + \nu$)
- ω -power (denoted $\mu \cdot \omega$)
- ω^* -power (denoted $\mu \cdot \omega^*$)
- shuffle (denoted $[\nu_1, \nu_2, \dots, \nu_k]^\eta$) for arbitrary $k \geq 1$.

The semantics of the concatenation, ω -power and ω^* -power generalize the corresponding operations for linear orders in the obvious way. To define the extended word $[\nu_1, \dots, \nu_k]^\eta$, let $\lambda: \mathbb{Q} \rightarrow \{1, 2, \dots, k\}$ be a mapping such that $\lambda^{-1}(i)$ is dense for all $1 \leq i \leq k$. Then set $\nu(q) = \nu_{\lambda(q)}$ for $q \in \mathbb{Q}$ and define

$$[\nu_1, \dots, \nu_k]^\eta = \sum_{q \in (\mathbb{Q}; \leq)} \nu(q)$$

as we did for linear orders. This operation is the extension of the shuffle sum from linear orders to extended words. It is well-defined in as far as the choice of the function λ does not influence the isomorphism type of the result. For a term t , let $|t|$ denote the extended word it describes.

Let $\nu = (L; \leq, \lambda)$ be an extended word. On the set L , we define an equivalence relation \sim by $x \sim y$ if (where we assume $x \leq y$)

- the interval $[x, y]$ is finite or
- for any $x', y', z \in [x, y]$ with $x' < y'$, there exists $z' \in (x', y')$ with $\lambda(z) = \lambda(z')$.

As explained above, regular words with a single \sim -equivalence class are of particular importance in our proof. These regular words can be described by “primitive terms in normal form” (consult [2, Definitions 57] for their formal definition). Here, we list only the main properties of the set $D(A)$ of all these primitive terms in normal form:

- The set $D(A)$ is decidable (clear by its definition).
- If ν is a regular word with a single \sim -equivalence class, then there exists a term $t \in D(A)$ with $\nu \cong |t|$ [2, Lemma 58].
- If $t \in D(A)$, then $|t|$ has a single \sim -equivalence class (clear by the definition of $D(A)$).
- If $s, t \in D(A)$ with $|s| \cong |t|$, then $s = t$ [2, Proposition 62].

Let $\nu = (L; \leq_{\text{lex}}, \lambda)$ be a regular word. The equivalence classes with respect to \sim are convex sets (or segments). Hence they can be ordered by

$$[x]_{\sim} <' [y]_{\sim} : \Longleftrightarrow x < y \text{ and } x \not\sim y$$

such that $(L/\sim; \leq')$ is a linear order. For $X \in L/\sim$, the restriction of ν to the equivalence class X is a regular word with a single \sim -equivalence class. Hence there exists a unique term $t_X \in D(A)$ with $|t| \cong \nu \upharpoonright X$. Define $\lambda': L/\sim \rightarrow D(A)$ by $X \mapsto t_X$. Then

$$c(\nu) = (L/\sim; \leq', \lambda')$$

is an extended word with possibly infinite alphabet.

► **Theorem 1.** *The set of rigid regular words ν (given as tuple of finite automata) is decidable.*

Proof. Let $\nu = (L; \leq_{\text{lex}}, \lambda)$ be a regular word given by finite automata that accept L and $\lambda^{-1}(a)$ for $a \in A$ (without loss of generality, we can assume $\varepsilon \notin L$). In a first step, we compute a term t with $|t| \cong \nu$ which is possible by Heilbrunner [15].

Using [2, Theorem 64], we construct a term $c(t)$ over $D(A)$ with $|c(t)| \cong c(|t|)$, in particular, $c(|t|)$ has a finite alphabet. From this term $c(t)$, we can extract the alphabet $D \subseteq D(A)$ of all symbols from $D(A)$ that appear in $c(t)$. Then we observe that $|t|$ has a nontrivial automorphism if and only if

- $c(|t|) = |c(t)|$ has a nontrivial automorphism or
- there exists a \sim -equivalence class X such that $|t| \upharpoonright X$ has a non-trivial automorphism.

Note that $s \in D$ if and only if there exists a \sim -equivalence class X with $|t| \upharpoonright X \cong |s|$. Hence the second item holds if and only if there exists $s \in D$ such that $|s|$ has a nontrivial automorphism – but this is the case if and only if s is of the form $u \cdot \omega^* + u \cdot \omega$ or $[u_1, \dots, u_k]^\eta$. To decide whether $|c(t)|$ has a nontrivial automorphism, we call this process recursively. From [22], we observe that $c^n(|t|)$ is a singleton for some $n \in \mathbb{N}$, hence this recursive procedure stops eventually with $|t|$ a singleton. ◀

► **Corollary 2.** *The set of regular languages L such that $(L; \leq_{\text{lex}})$ is rigid, is decidable.*

3.2 Regular universe and \leq_{lex}^2

The situation changes completely when we move from the lexicographic order \leq_{lex} to the linear order \leq_{lex}^2 since, as we will see, rigidity of $(L; \leq_{\text{lex}}^2)$ is undecidable for regular languages L .

Let $p, q \in \mathbb{N}[\bar{x}]$ be two polynomials with coefficients in \mathbb{N} and variables among $\bar{x} = (x_1, \dots, x_k)$. Then define the scattered linear order

$$\mathcal{L}_{p,q} = \sum_{\bar{x} \in (\mathbb{N}^k; \leq_{\text{lex}})} \left((\underline{p(\bar{x})} + \delta) \cdot \omega^* + (\underline{q(\bar{x})} + \delta) \cdot \omega \right).$$

This linear order $\mathcal{L}_{p,q}$ forms an ω -sequence of “blocks” of the form

$$B(m, n) = (\underline{m} + \delta) \cdot \omega^* + (\underline{n} + \delta) \cdot \omega$$

with $m, n \in \mathbb{N}$. Therefore, every automorphism of $\mathcal{L}_{p,q}$ has to map every block onto itself. In other words, $\mathcal{L}_{p,q}$ is rigid if and only if all these blocks are rigid. But $B(m, n)$ is rigid if and only if $m \neq n$. Hence we showed

$$\mathcal{L}_{p,q} \text{ is rigid} \iff \forall \bar{x} \in \mathbb{N}^k: p(\bar{x}) \neq q(\bar{x}).$$

We now prove that $\mathcal{L}_{p,q}$ is word-automatic or, more specifically, we will construct a regular set $L \subseteq \{0, 1\}^+ \otimes \{0, 1\}^+$ such that $\mathcal{L}_{p,q} \cong (L; \leq_{\text{lex}}^2)$ (see Lemma 4 below).

Let $\mathcal{A} = (Q, \iota, \Delta, F)$ be a word automaton over the alphabet Σ and let $w \in \Sigma^+$ be a word. Then $\text{Run}(\mathcal{A}, w)$ is the set of all words over Δ of the form

$$(q_0, a_1, q_1, \iota)(q_1, a_2, q_2, \iota) \dots (q_{k-1}, a_k, \iota, \iota)$$

with $w = a_1 a_2 \dots a_k$ and $q_0 \in F$. These words encode the accepting runs of the word automaton \mathcal{A} (recall that word automata are special bottom up tree automata which explains the unusual position of the initial and final states in the run). Furthermore, let $\text{Run}(\mathcal{A}) = \bigcup_{w \in \Sigma^+} \text{Run}(\mathcal{A}, w)$.

► **Lemma 3.** *From polynomials $p, q \in \mathbb{N}[x_1, \dots, x_k]$, one can construct an alphabet Σ and a regular language $K \subseteq \Sigma^+ \otimes \Sigma^+$ such that $(K; \leq_{\text{lex}}^2) \cong \mathcal{L}_{p,q}$.*

If $\mathcal{L}_{p,q}$ has a non-trivial automorphism, then $(K; \leq_{\text{lex}}^2)$ has a non-trivial automatic automorphism.

Proof. Let p and q be polynomials from $\mathbb{N}[x_1, \dots, x_k]$. For $\bar{x} = (x_1, \dots, x_k) \in \mathbb{N}^k$, set

$$a^{\bar{x}} = a^{x_1} \zeta a^{x_2} \zeta \dots \zeta a^{x_k} \zeta \in (a^* \zeta)^k.$$

Then, as in the proof of [24, Lemma 7], one can construct nondeterministic finite automata $\mathcal{A}_p = (Q_p, \iota_p, \Delta_p, F_p)$ and $\mathcal{A}_q = (Q_q, \iota_q, \Delta_q, F_q)$ with $L(\mathcal{A}_p), L(\mathcal{A}_q) \subseteq (a^* \zeta)^k$, such that, for $\bar{x} \in \mathbb{N}^k$, the NFA \mathcal{A}_p has precisely $p(\bar{x})$ many accepting runs on the word $a^{\bar{x}}$, i.e., $|\text{Run}(\mathcal{A}_p, a^{\bar{x}})| = p(\bar{x})$, and similarly $|\text{Run}(\mathcal{A}_q, a^{\bar{x}})| = q(\bar{x})$. We will assume $\Delta_p \cap \Delta_q = \emptyset$.

Let $\Sigma = \{a, \zeta, 0, 1, 2, 3\} \cup \Delta_p \cup \Delta_q$ and let $K \subseteq \Sigma^+ \otimes \Sigma^+$ be the union (for $\bar{x} \in \mathbb{N}^k$) of the languages

$$\left(a^{\bar{x}} 0^+ 1 \otimes (\text{Run}(\mathcal{A}_p, a^{\bar{x}}) \cup 3 2^+ 3^+ 2) \right) \cup \left(a^{\bar{x}} 1^+ 0 \otimes (\text{Run}(\mathcal{A}_q, a^{\bar{x}}) \cup 3 2^+ 3^+ 2) \right).$$

We have to show that the language K is effectively regular. Here, the crucial point is the regularity of

$$\bigcup_{\bar{x} \in \mathbb{N}^k} a^{\bar{x}} 0^+ 1 \otimes \text{Run}(\mathcal{A}_p, a^{\bar{x}}) = \left[\bigcup_{\bar{x} \in \mathbb{N}^k} a^{\bar{x}} \otimes \text{Run}(\mathcal{A}_p, a^{\bar{x}}) \right] \cdot (0^+ 1 \otimes \{\varepsilon\})$$

(this equality holds since $|w| = |W|$ for any $w \in (a^* \dot{c})^k$ and $W \in \text{Run}(\mathcal{A}_p, w)$). But a word belongs to the language in square brackets if and only if it is the convolution of a word w from the regular language $(a^* \dot{c})^k$ and a run of the automaton \mathcal{A}_p on this word w , a property that a finite automaton can check easily.

On the alphabet Σ , we fix a linear order \leq such that $\Delta_p \cup \Delta_q < 0 < 1 < 2 < 3 < \dot{c} < a$. Then $(K; \leq_{\text{lex}}^2) \cong \mathcal{L}_{p,q}$.

Now suppose that $\mathcal{L}_{p,q}$ has a non-trivial automorphism. Then, as we saw above, there is $\bar{y} \in \mathbb{N}^k$ such that $p(\bar{y}) = q(\bar{y})$. Then $\mathcal{L}_{p,q}$ contains an interval isomorphic to a \mathbb{Z} -sequence of copies of $p(\bar{y}) + \delta \cong q(\bar{y}) + \delta$. Moving these blocks in $(K; \leq_{\text{lex}}^2)$ upwards by 1 and fixing everything else in K gives a non-trivial automatic automorphism. \blacktriangleleft

► **Lemma 4.** *From polynomials $p, q \in \mathbb{N}[x_1, \dots, x_k]$, one can construct a regular language $L \subseteq \{0, 1\}^+ \otimes \{0, 1\}^+$ such that $(L; \leq_{\text{lex}}^2) \cong \mathcal{L}_{p,q}$.*

If $\mathcal{L}_{p,q}$ has a non-trivial automorphism, then $(L; \leq_{\text{lex}}^2)$ has a non-trivial automatic automorphism.

Proof. Let $p, q \in \mathbb{N}[x_1, \dots, x_k]$ be polynomials, let Σ be the alphabet and K the language from Lemma 3, and let $(\Sigma; \leq)$ be the sequence $\sigma_1 < \sigma_2 < \dots < \sigma_\ell$. Furthermore, let g denote the monoid homomorphism from Σ^* to $\{0, 1\}^*$ defined by $g(\sigma_i) = 1^i 0^{\ell-i}$ for $1 \leq i \leq \ell$. Now set $L = \{g(u) \otimes g(v) \mid u \otimes v \in K\}$. Then $u \otimes v \mapsto g(u) \otimes g(v)$ is an isomorphism from $(K; \leq_{\text{lex}}^2)$ onto $(L; \leq_{\text{lex}}^2)$. Since all the words $g(\sigma_i)$ have the same length, the language L is also regular. \blacktriangleleft

The set of pairs of polynomials $p, q \in \mathbb{N}[\bar{x}]$ with $p(\bar{y}) \neq q(\bar{y})$ for all $\bar{y} \in \mathbb{N}^k$ is Π_1^0 -complete [26]. This allows to prove the following result:

- **Theorem 5.** (i) *The set of regular languages $L \subseteq \{0, 1\}^+ \otimes \{0, 1\}^+$ such that $(L; \leq_{\text{lex}}^2)$ is rigid (is rigid and scattered, resp.), is Π_1^0 -hard.*
(ii) *The set of regular languages $L \subseteq \{0, 1\}^+ \otimes \{0, 1\}^+$ such that $(L; \leq_{\text{lex}}^2)$ is automatically rigid (automatically rigid and scattered, resp.) is Π_1^0 -hard.*

- **Corollary 6.** (i) *The set of word-automatic presentations of rigid (rigid and scattered, resp.) linear orders is Π_1^0 -hard.*
(ii) *The set of word-automatic presentations of automatically rigid (automatically rigid and scattered, resp.) linear orders is Π_1^0 -complete.*

3.3 Context-free universe and \leq_{lex}

Ésik initiated the investigation of linear orders of the form $(L; \leq_{\text{lex}})$ where L is context-free. Density of such a linear order is undecidable [14], the isomorphism problem is Σ_1^1 -complete [24] and their rank is bounded by ω^ω [7].

We will show that rigidity of $(L; \leq_{\text{lex}})$ is undecidable for deterministic context-free languages L . The proof uses the linear order $\mathcal{L}_{p,q}$ and constructs a deterministic context-free language L' such that $(L'; \leq_{\text{lex}}) \cong \mathcal{L}_{p,q}$. This construction is a variant of the construction in the proof of Lemma 3.

► **Lemma 7.** *From polynomials $p, q \in \mathbb{N}[x_1, \dots, x_k]$, one can construct a deterministic context-free language $L' \subseteq \{0, 1\}^+$ such that $(L'; \leq_{\text{lex}}) \cong \mathcal{L}_{p,q}$.*

Proof. Let $p, q \in \mathbb{N}[x_1, \dots, x_k]$ be polynomials, let Σ be the alphabet and let K be the language from Lemma 3. Then set

$$K' = \{u\$v^{rev} \mid u \otimes v \in K\}$$

where v^{rev} is the reversal of the word v . Then, from a deterministic finite automaton \mathcal{A} accepting K^{rev} , one can construct a deterministic pushdown automaton accepting K' . Recall that there is a word u such that $u \otimes 3232 <_{\text{lex}}^2 u \otimes 32232$ both belong to K . But we have $u\$2323 >_{\text{lex}} u\23223 (the same phenomenon can be observed with words $u \otimes \rho$ where ρ is a run of one of the two weighted automata). In other words, the obvious mapping $u \otimes v \mapsto u\$v^{rev}$ is no isomorphism from $(K; \leq_{\text{lex}}^2)$ onto $(K'; \leq_{\text{lex}})$.

Note that the alphabet of K' is $\Sigma' = \{\$ \} \cup \Sigma = \{\$, a, \zeta, 0, 1, 2, 3\} \cup \Delta_p \cup \Delta_q$. We order Σ' in such a way that $\Delta_p \cup \Delta_q < 0 < 1 < 3 < 2 < \zeta < a < \$$. Compared to the proof of Lemma 3, the order of 2 and 3 is inverted and $\$$ is made the new maximal element. The reason for this inversion is that now, we have $(32^+3^+2^{rev}; \leq'_{\text{lex}}) \cong (32^+3^+2; \leq_{\text{lex}}) \cong \delta$. Given this definition and observation, one can show $(K'; \leq'_{\text{lex}}) \cong (K; \leq_{\text{lex}}^2)$ which was isomorphic to $\mathcal{L}_{p,q}$. The construction of $L' \subseteq \{0, 1\}^+$ then follows the proof of Lemma 4. ◀

Now we obtain, in the same way that we proved Theorem 5, the following result.

► **Theorem 8.** *The set of context-free languages $L \subseteq \{0, 1\}^+$ such that $(L; \leq_{\text{lex}})$ is rigid (is rigid and scattered, resp.), is Π_1^0 -hard.*

4 Isomorphisms and automorphisms of linear orders on tree languages

In this section, we will show that the isomorphism of scattered and tree-automatic linear orders is undecidable. Furthermore, we will prove that the existence of a non-trivial automorphism in this case is Σ_2^0 -hard. Both these results use (an improved version of) a theorem by Krob [23] from [10] that we discuss first.

4.1 Weighted automata and Minsky machines

A *weighted automaton* is a tuple $\mathcal{A} = (Q, \Sigma, \iota, \mu, F)$ where Q is the finite set of states, Σ is the alphabet, $\iota \in Q$ is the initial state, $F \subseteq Q$ is the set of accepting states, and $\mu: Q \times \Sigma \times Q \rightarrow \{-\infty, 0, 1\}$ is the weight function.

A *run* of \mathcal{A} is a sequence $\rho = (q_0, a_1, q_1) \dots (q_{k-1}, a_k, q_k) \in \Delta^+$ such that $q_0 = \iota$, $\mu(q_{i-1}, a_i, q_i) \in \{0, 1\}$ for all $1 \leq i \leq k$, and $q_k \in F$. Its *label* is the word $a_1 \dots a_k \in \Sigma^+$ and its *weight* $\text{wt}(\rho)$ is the number of indices $i \in \{1, 2, \dots, k\}$ with $\mu(q_{i-1}, a_i, q_i) = 1$. By $\text{Run}(\mathcal{A}, w)$ we denote the set of runs labeled w and $\text{Run}(\mathcal{A})$ denotes the set of all runs of \mathcal{A} . The *behaviour* $\|\mathcal{A}\|$ of \mathcal{A} is the function from Σ^+ to $\mathbb{N} \cup \{-\infty\}$ that maps the word w to the maximal weight of a run with label w .

► **Theorem 9** (cf. proof of [10, Theorem 8.6]). *From a Minsky machine (or two-counter automaton) \mathcal{M} , one can construct a weighted automaton \mathcal{A} and a regular language $\text{CT}_{\text{reg}} \subseteq (\Sigma \cdot \square)^+$ such that, for any $m \in \mathbb{N}$, the following are equivalent:*

1. m is not accepted by \mathcal{M} .
2. $\|\mathcal{A}\|(u) > \frac{1}{2}|u|$ for all $u \in \text{CT}_{\text{reg}}$ with $m = \max\{n \mid \$\square(a\square)^n \leq_{\text{pref}} u\}$.

Furthermore, $\|\mathcal{A}\|(u) \in \mathbb{N}$ for all $u \in \text{CT}_{\text{reg}}$.

From the weighted automaton \mathcal{A} , one can then construct (cf. [9, 10]) weighted automata $\mathcal{A}_{\mathcal{M}}$ on the alphabet Σ and $\mathcal{B}_{\mathcal{M}}$ on the alphabet $\Sigma_{\#}^2$ such that

$$\begin{aligned} \|\mathcal{A}_{\mathcal{M}}\|(u) &= \max(\lfloor \frac{|u|}{2} \rfloor + 1, \|\mathcal{A}\|(u)) \text{ and} \\ \|\mathcal{B}_{\mathcal{M}}\|(x) &= \begin{cases} \|\mathcal{A}\|(u) & \text{if } x = u \otimes \$\square(a\square)^m, u \in \text{CT}_{\text{reg}}, \\ & \text{and } m = \max\{n \mid \$\square(a\square)^n \leq_{\text{pref}} u\} \\ \|\mathcal{A}_{\mathcal{M}}\|(u) & \text{if } x = u \otimes \$\square(a\square)^m \text{ and} \\ & (u \notin \text{CT}_{\text{reg}} \text{ or } m \neq \max\{n \mid \$\square(a\square)^n \leq_{\text{pref}} u\}) \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

for all $u \in \Sigma^+$ and $x \in (\Sigma_{\#}^2)^+$.

For $m \in \mathbb{N}$, we define the function $r_{\mathcal{M},m}: \Sigma^+ \rightarrow \mathbb{N}$ by $r_{\mathcal{M},m}(u) = \|\mathcal{B}_{\mathcal{M}}\|(u \otimes \$\square(a\square)^m)$. This is well-defined since, for any $u \in \Sigma^+$ and $m \in \mathbb{N}$, we have $\|\mathcal{A}\|(u) \in \mathbb{N}$ and therefore also $\|\mathcal{B}_{\mathcal{M}}\|(u \otimes \$\square(a\square)^m) \in \mathbb{N}$. In other words, we have

$$r_{\mathcal{M},m}(u) = \begin{cases} \|\mathcal{A}\|(u) & \text{if } u \in \text{CT}_{\text{reg}} \text{ and } m = \max\{n \mid \$\square(a\square)^n \leq_{\text{pref}} u\} \\ \|\mathcal{A}_{\mathcal{M}}\|(u) & \text{otherwise.} \end{cases}$$

The following is the central property from this section that we will use in our handling of tree-automatic linear orders.

► **Proposition 10.** For all $m \in \mathbb{N}$, the following are equivalent:

1. m is not accepted by the Minsky machine \mathcal{M} .
2. $\|\mathcal{A}_{\mathcal{M}}\|(u) = r_{\mathcal{M},m}(u)$ holds for all $u \in \Sigma^*$.

► **Remark.** Fix some Minsky machines \mathcal{M} that accepts an undecidable set of natural numbers. From $m \in \mathbb{N}$, one can then construct a weighted automaton \mathcal{B} with $\|\mathcal{B}\| = r_{\mathcal{M},m}$. Hence the set of weighted automata \mathcal{B} with $\|\mathcal{B}\| = \|\mathcal{A}_{\mathcal{M}}\|$ is Π_1^0 -hard and therefore undecidable. This strengthens Krob's result stating that the set of pairs $(\mathcal{A}, \mathcal{B})$ of weighted automata with $\|\mathcal{A}\| = \|\mathcal{B}\|$ is Π_1^0 -hard.

4.2 Isomorphism

Note that Krob's result talks about functions $\Sigma^+ \rightarrow \mathbb{N}$ while we are interested in linear orders. Therefore, we set

$$\mathcal{L}_r = \sum_{w \in (\Sigma^+; \leq_{\text{lex}})} (\omega^{r(w)+1} + \delta)$$

for a function $r: \Sigma^+ \rightarrow \mathbb{N}$. Since $(\Sigma^+; \leq_{\text{lex}}) \cong \omega$, this linear order is an ω -sequence of ordinals of the form ω^n with $n \geq 1$, separated by our delimiter δ . Hence it is scattered. Furthermore, for all functions $r, r': \Sigma^+ \rightarrow \mathbb{N}$, we obtain

$$\mathcal{L}_r \cong \mathcal{L}_{r'} \iff r = r'. \quad (1)$$

The following lemma states in particular that \mathcal{L}_r is tree automatic whenever $r = \|\mathcal{A}\|$ for some weighted automaton \mathcal{A} .

► **Lemma 11.** From a weighted automaton \mathcal{A} , one can compute a regular set of trees $L_{\mathcal{A}}$ such that $(L_{\mathcal{A}}; \leq_{\text{trees}}) \cong \mathcal{L}_{\|\mathcal{A}\|}$.

Before we prove this lemma, we show how we can use it to prove that the isomorphism problem of scattered tree-automatic linear orders is undecidable (the proof of Lemma 11 can be found following the proof of Corollary 14).

► **Lemma 12.** *From a Minsky machine \mathcal{M} and $m \in \mathbb{N}$, one can compute a regular set of trees L such that $(L; \leq_{\text{trees}}) \cong \mathcal{L}_{r_{\mathcal{M},m}}$.*

Proof. Let \mathcal{M} be a Minsky machine and let $m \in \mathbb{N}$. Let $\mathcal{B}_{\mathcal{M}}$ be the weighted automaton constructed following Theorem 9. Then, from $m \in \mathbb{N}$, we can compute a weighted automaton $\mathcal{B}_{\mathcal{M},m}$ with alphabet Σ such that

$$\|\mathcal{B}_{\mathcal{M},m}\|(u) = \|\mathcal{B}_{\mathcal{M}}\|(u \otimes \$\square(a\square)^m) \text{ for all } u \in \Sigma^+.$$

But then $\|\mathcal{B}_{\mathcal{M},m}\| = r_{\mathcal{M},m}$. By Lemma 11, we can compute, from $m \in \mathbb{N}$, a regular language of trees L such that $(L; \leq_{\text{trees}}) \cong \mathcal{L}_{\|\mathcal{B}_{\mathcal{M},m}\|} = \mathcal{L}_{r_{\mathcal{M},m}}$. ◀

► **Theorem 13.** *There is a scattered linear order \mathcal{L} such that the set of regular tree languages L with $(L; \leq_{\text{trees}}) \cong \mathcal{L}$ is Π_1^0 -hard.*

Proof. Let $P \subseteq \mathbb{N}$ be some Π_1^0 -complete set. Then there exists a Minsky machine \mathcal{M} that accepts the set $\mathbb{N} \setminus P$. Let $\mathcal{A}_{\mathcal{M}}$ and $\mathcal{B}_{\mathcal{M}}$ be the weighted automata constructed following Theorem 9. Then we get

$$\begin{aligned} m \in P &\iff m \text{ is not accepted by } \mathcal{M} \\ &\stackrel{\text{Prop. 10}}{\iff} \|\mathcal{A}_{\mathcal{M}}\|(u) = r_{\mathcal{M},m}(u) \text{ for all } u \in \Sigma^+ \\ &\iff \mathcal{L}_{\|\mathcal{A}_{\mathcal{M}}\|} \cong \mathcal{L}_{r_{\mathcal{M},m}} \end{aligned}$$

where the last equivalence follows from (1). Hence, by Lemma 12, we can reduce the Π_1^0 -complete set P to the set of regular tree languages L with $(L; \leq_{\text{trees}}) \cong \mathcal{L}_{\|\mathcal{A}_{\mathcal{M}}\|}$. The theorem therefore holds with $\mathcal{L} = \mathcal{L}_{\|\mathcal{A}_{\mathcal{M}}\|}$. ◀

Since the linear order \leq_{trees} is tree-automatic, we immediately obtain

► **Corollary 14.** *There is a scattered linear order \mathcal{L} whose set of tree-automatic presentations is Π_1^0 -hard.*

One immediately gets that the isomorphism problem for tree-automatic scattered linear orders is Π_1^0 -hard. We do not know whether the set of tree-automatic presentations of *scattered* linear orders is decidable. Therefore, the following immediate consequence of Corollary 14 is a bit stronger:

► **Corollary 15.** *Let X be a set of pairs of tree-automatic presentations such that, for all tree-automatic presentations P_1 and P_2 of scattered linear orders \mathcal{L}_1 and \mathcal{L}_2 , one has*

$$(P_1, P_2) \in X \iff \mathcal{L}_1 \cong \mathcal{L}_2.$$

Then X is Π_1^0 -hard.

The rest of this section is devoted to the proof of Lemma 11.

Proof of Lemma 11. Let $\mathcal{A} = (Q, \Sigma, \iota, \mu, F)$ be a weighted automaton. We will construct a tree-automatic presentation of the linear order $\mathcal{L}_{\|\mathcal{A}\|}$.

A *run tree* of \mathcal{A} is a tree t over the alphabet $\Sigma \uplus \{\$ \}$ such that there exist states $\iota = q_0, q_1, \dots, q_{k-1} \in Q$ and $q_k \in F$ (with $k = \max\{i \mid 0^{i+1} \in \text{dom}(t)\}$) with the following properties (see the tree on the next page with $k = 5$ where we omitted the label $\$$):

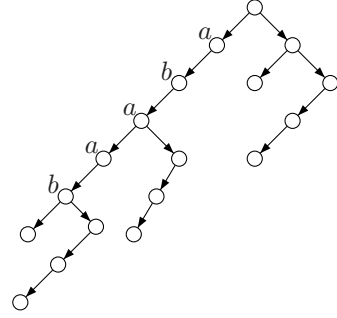
- (T1) $11 \in \text{dom}(t) \subseteq 0^* \cup 0^*10^* \cup 110^*$ and $100 \notin \text{dom}(t)$
 (T2) $t(0^i) \in \Sigma$ and $\mu(q_{i-1}, t(0^i), q_i) \neq -\infty$ for all $1 \leq i \leq k$
 (T3) $0^i1 \in \text{dom}(t)$ implies $1 \leq i \leq k$ and $\mu(q_{i-1}, a_i, q_i) = 1$ or $i = 0$
 (T4) $t^{-1}(\$) = \text{dom}(t) \setminus \{0^i \mid 1 \leq i \leq k\}$

Note that every run tree t defines a word over Σ , namely

$$\text{word}(t) = t(0) t(00) \dots t(0^k).$$

Since $11 \in \text{dom}(t)$, also 10 and therefore 1 belong to $\text{dom}(t)$. Since, consequently, $0 \in \text{dom}(t)$, we have $\text{word}(t) \neq \varepsilon$.

The figure on the right shows a run tree t with $\text{word}(t) = abaab$. The idea is that the “main branch” $\{0, 00, \dots, 0^k\}$ carries a run ρ of the weighted automaton \mathcal{A} . The number of “side branches” starting in some node 0^i1 with $i > 0$ is at most the weight $\text{wt}(\rho)$ of the encoded run. Since these side branches have arbitrary length, the whole run tree stands for an element of $\omega^{\text{wt}(\rho)}$. The “side branch” starting in 11 plays a special role, its length $|\text{dom}(t) \cap 110^+|$ is denoted $n(t)$ (the run tree t on the right satisfies $n(t) = 2$).



We next define, for two trees s and t , the tree $s + t$ by adding a new $\$$ -labeled root and considering s as left subtree of $s + t$ and t as right subtree. In particular, we will need trees of the form $w + t$ with w a word. These trees carry the sequences $\$w$ on $\text{dom}(w + t) \cap 0^*$ and satisfy $(w + t) \upharpoonright_1 \cong t$.

We consider the language $L_{\mathcal{A}} = \{t \mid t \text{ is a run tree}\} \cup \{w\$ + t \mid w \in \Sigma^+, t \in D\}$ where D is the set of trees from page 458 that satisfies $(D; \leq_{\text{trees}}) \cong \delta$. This language $L_{\mathcal{A}}$ is regular.

Note that trees from $L_{\mathcal{A}}$ use the alphabet $\Sigma \cup \{\$\}$ that we order arbitrarily. We will now prove $(L_{\mathcal{A}}; \leq_{\text{trees}}) \cong \mathcal{L}_{\|\mathcal{A}\|}$.

First let $w \in \Sigma^+$ and $n \in \mathbb{N}$. Then let $\mathcal{I}_{w,n}^1$ denote the restriction of $(L_{\mathcal{A}}; \leq_{\text{trees}})$ to all run trees t with

$$\text{word}(t) = w \text{ and } n(t) = n. \quad (2)$$

Note that for any two run trees s and t satisfying (2), we have $s \upharpoonright_{0^*} = t \upharpoonright_{0^*}$ and $s \upharpoonright_1 = t \upharpoonright_1$. Hence $s \leq_{\text{trees}} t$ if and only if there exists $i \geq 1$ with $t \upharpoonright_{0^j1} = s \upharpoonright_{0^j1}$ for all $1 \leq j < i$ and $s \upharpoonright_{0^i1} <_{\text{trees}} t \upharpoonright_{0^i1}$. By (T3), $\text{dom}(t) \cap 0^+1$ contains at most $|w|$ elements. Furthermore note that the trees $t \upharpoonright_{0^i1}$ can be identified with natural numbers (namely with $|\text{dom}(t) \cap 0^i10^*|$). This shows that $\mathcal{I}_{w,n}^1$ can be embedded into $(\mathbb{N}^{|w|}; \leq_{\text{lex}})$ and is therefore well-ordered and at most $\omega^{|w|}$.

Now let $\rho = (q_0, a_1, q_1)(q_1, a_2, q_2) \dots (q_{k-1}, a_k, q_k) \in \text{Run}(\mathcal{A}, w)$ be a run of the weighted automaton \mathcal{A} on the word $w = a_1 \dots a_k$. Consider a tuple $(m_1, \dots, m_k) \in \mathbb{N}^k$ such that $m_i > 0$ implies $\mu(q_{i-1}, a_i, q_i) = 1$. Then there exists a unique run tree t satisfying (2) and $|\text{dom}(t) \cap 0^i10^*| = m_i$ for all $1 \leq i \leq k$. This gives an order-preserving embedding $f_\rho: \omega^{\text{wt}(\rho)} \rightarrow \mathcal{I}_{w,n}^1$, i.e., we showed $\omega^{\text{wt}(\rho)} \leq \mathcal{I}_{w,n}^1$. Since this holds for arbitrary runs $\rho \in \text{Run}(\mathcal{A}, w)$ and since $\|\mathcal{A}\|(w) = \max\{\text{wt}(\rho) \mid \rho \in \text{Run}(\mathcal{A}, w)\}$, we get $\omega^{\|\mathcal{A}\|(w)} \leq \mathcal{I}_{w,n}^1$ and therefore $\omega^{\|\mathcal{A}\|(w)+1} \leq \mathcal{I}_{w,n}^1 \cdot \omega$.

By (T2), for every run tree t satisfying (2), there exists at least one run $\rho \in \text{Run}(\mathcal{A}, w)$ such that t is in the image of the embedding f_ρ . Hence $\mathcal{I}_{w,n}^1 \leq \bigoplus_{\rho \in \text{Run}(\mathcal{A}, w)} \omega^{\text{wt}(\rho)}$ where

\oplus denotes the natural sum of ordinals. We can conclude

$$\begin{aligned}\omega^{||\mathcal{A}||(w)+1} &\leq \mathcal{I}_{w,n}^1 \cdot \omega \leq \left(\bigoplus_{\rho \in \text{Run}(\mathcal{A},w)} \omega^{\text{wt}(\rho)} \right) \cdot \omega \\ &= \omega^{\max\{\text{wt}(\rho) \mid \rho \in \text{Run}(\mathcal{A},w)\}+1} \\ &= \omega^{||\mathcal{A}||(w)+1}\end{aligned}$$

and therefore $\mathcal{I}_{w,n}^1 \cdot \omega = \omega^{||\mathcal{A}||(w)+1}$.

Next consider the restriction \mathcal{I}_w^1 of $(L_{\mathcal{A}}; \leq_{\text{trees}})$ to the set of run trees t with $\text{word}(t) = w$. Then $n(s) < n(t)$ implies $s <_{\text{trees}} t$. Furthermore, the restriction of \mathcal{I}_w^1 to the set of run trees t with $n(t) = n$ equals $\mathcal{I}_{w,n}^1$. Hence $\mathcal{I}_w^1 = \sum_{n \in (\mathbb{N}; \leq)} \mathcal{I}_{w,n}^1 = \mathcal{I}_{w,0}^1 \cdot \omega = \omega^{||\mathcal{A}||(w)+1}$ since $\mathcal{I}_{w,0}^1 \cong \mathcal{I}_{w,n}^1$ for all $n \geq 0$.

Next consider the restriction \mathcal{I}_w^2 of $(L_{\mathcal{A}}; \leq_{\text{trees}}^2)$ to the set of trees $w\$ + D$. Then $\mathcal{I}_w^2 \cong \delta$ by what we saw on page 458. Let s be a run tree with $\text{word}(s) = w$ and let $t \in w\$ + D$. Then s and t coincide on 0^* (where they both carry the sequence $\$w\$$). Consider $s \upharpoonright_{10^*}$ and $t \upharpoonright_{10^*}$. Since s is a run tree, we have $\text{dom}(s) \cap 10^* = \{1, 10\}$ while $t \upharpoonright_1 \in D$ implies $\text{dom}(t) \cap 10^* = \{1, 10, 100\}$. Hence $s \upharpoonright_1 <_{\text{trees}} t \upharpoonright_1$ and therefore $s <_{\text{trees}} t$. Hence, the restriction \mathcal{I}_w of $(L_{\mathcal{A}}; \leq_{\text{trees}})$ to the set of run trees t with $\text{word}(t) = w$ and the set of trees $w\$ + D$ satisfies $\mathcal{I}_w = \mathcal{I}_w^1 + \mathcal{I}_w^2 \cong \omega^{||\mathcal{A}||(w)+1} + \delta$.

Finally, let $u, v \in \Sigma^+$. Then $u \leq_{\text{lex}} v$ if and only if $u \leq_{\text{trees}} v$. This implies

$$(L_{\mathcal{A}}; \leq_{\text{trees}}) = \sum_{w \in (\Sigma^+; \leq_{\text{lex}})} \mathcal{I}_w \cong \sum_{w \in (\Sigma^+; \leq_{\text{lex}})} \omega^{||\mathcal{A}||(w)+1} + \delta = \mathcal{L}_{\mathcal{A}}.$$

◀

4.3 Automorphisms

From Theorem 5, we already know that the existence of a non-trivial automorphism of a word-automatic and scattered linear order is Σ_1^0 -hard. Here, we push this lower bound one level higher for tree-automatic scattered linear orders. The order theoretic construction resembles that from Section 3.2, but also uses ideas from the previous section.

Let \mathcal{M} be a Minsky machine, let $\mathcal{A}_{\mathcal{M}}$ and $\mathcal{B}_{\mathcal{M}}$ be the weighted automata and, for $m \in \mathbb{N}$, let $r_{\mathcal{M},m}$ be the function defined following Theorem 9. Then we define the linear order

$$\mathcal{L}_{\mathcal{M}} = \sum_{m \in (\mathbb{N}; \leq)} (\mathcal{L}_{||\mathcal{A}_{\mathcal{M}}||} \cdot \omega^* + \mathcal{L}_{r_{\mathcal{M},m}} \cdot \omega).$$

Note that this linear order is rigid if and only if $\mathcal{L}_{||\mathcal{A}_{\mathcal{M}}||} \not\cong \mathcal{L}_{r_{\mathcal{M},m}}$ for all $m \in \mathbb{N}$. But this is the case if and only if \mathcal{M} accepts all natural numbers m , a Π_2^0 -complete problem.

► **Lemma 16.** *From a Minsky machine \mathcal{M} , one can construct a tree-automatic presentation of the linear order $\mathcal{L}_{\mathcal{M}}$.*

Proof. Let \mathcal{M} be a Minsky machine, let $\mathcal{A}_{\mathcal{M}}$ and $\mathcal{B}_{\mathcal{M}}$ be the weighted automata and let $r_{\mathcal{M},m}: \Sigma^+ \rightarrow \mathbb{N}$ be the function defined following Theorem 9. Recall that the alphabet of $\mathcal{A}_{\mathcal{M}}$ is Σ and that of $\mathcal{B}_{\mathcal{M}}$ is $\Sigma_{\#}^2$. Recall the notion of a run tree from the proof of Lemma 11 that is based on a weighted automaton. In this proof, we will consider run trees with respect to the weighted automaton $\mathcal{A}_{\mathcal{M}}$ and with respect to the weighted automaton $\mathcal{B}_{\mathcal{M}}$. Now recall the definition of the language $L_{\mathcal{A}_{\mathcal{M}}}$ and $L_{\mathcal{B}_{\mathcal{M}}}$:

$$\begin{aligned}L_{\mathcal{A}_{\mathcal{M}}} &= \{t \mid t \text{ is a run tree wrt. } \mathcal{A}_{\mathcal{M}}\} \cup \{w\$ + t \mid w \in \Sigma^+, t \in D\} \\ L_{\mathcal{B}_{\mathcal{M}}} &= \{t \mid t \text{ is a run tree wrt. } \mathcal{B}_{\mathcal{M}}\} \cup \{w\$ + t \mid w \in (\Sigma_{\#}^2)^+, t \in D\}\end{aligned}$$

Note that these two tree languages are disjoint since the alphabets Σ and $\Sigma_{\#}^2$ are disjoint. Now define the language

$$\begin{aligned} L_{\mathcal{M}} = & (L_{\mathcal{A}\mathcal{M}} \otimes \$^* \otimes \$\square(a\square)^*) \\ & \cup \{t \otimes \$^k \otimes \$\square(a\square)^m \mid k, m \in \mathbb{N}, t \in L_{\mathcal{B}\mathcal{M}}, \text{ and} \\ & (t \text{ is a run tree} \Rightarrow \text{word}(t) \in \Sigma^+ \otimes \$\square(a\square)^m)\}. \end{aligned}$$

The crucial point regarding the regularity of this set is the verification that a tree $t \otimes \$^k \otimes \$\square(a\square)^m$ with t a run tree of $\mathcal{B}_{\mathcal{M}}$ belongs to the second set. But this is the case if $t|_{0^*} = \$\square(a\square)^m\$$, a property that a tree automaton can check easily.

On this set, we define the following linear order \preceq : $(s \otimes \$^k \otimes \$\square(a\square)^m) \preceq (t \otimes \$^\ell \otimes \$\square(a\square)^n)$ if and only if one of the following hold

- (O1) $m < n$ or
- (O2) $m = n$, $s \in L_{\mathcal{A}\mathcal{M}}$, and $t \in L_{\mathcal{B}\mathcal{M}}$, or
- (O3) $m = n$, $s, t \in L_{\mathcal{A}\mathcal{M}}$, and $k > \ell$, or
- (O4) $m = n$, $s, t \in L_{\mathcal{A}\mathcal{M}}$, $k = \ell$, and $s \leq_{\text{trees}} t$, or
- (O5) $m = n$, $s, t \in L_{\mathcal{B}\mathcal{M}}$, and $k < \ell$, or
- (O6) $m = n$, $s, t \in L_{\mathcal{B}\mathcal{M}}$, $k = \ell$, and $s \leq_{\text{trees}} t$.

It is clear that this relation is automatic and one can show $(L_{\mathcal{M}}; \preceq) \cong \mathcal{L}_{\mathcal{M}}$. ◀

- **Theorem 17.** (i) *The set of tree-automatic presentations of rigid (rigid and scattered, resp.) linear orders is Π_2^0 -hard.*
- (ii) *The set of tree-automatic presentations of automatically rigid linear orders is Π_1^0 -complete.*

5 Open questions

The isomorphism and rigidity problems for word-automatic scattered linear orders both belong to Δ_ω^0 (cf. [24]), our lower bound Π_1^0 for the rigidity problem leaves quite some room for improvements. Since the rank of a tree-automatic linear order is properly below ω^ω [18, 17], the proof of [24] can be adapted to show that the isomorphism and the rigidity problems for tree-automatic scattered linear orders both belong to $\Sigma_{\omega^\omega}^0$. But we only have the lower bounds Π_1^0 and Π_2^0 , resp. Finally, the rigidity problem for arbitrary word or tree-automatic linear orders is in Π_1^1 , but also here, we only have the arithmetic lower bound Π_1^0 and Π_2^0 , resp.

But the most pressing open question is the isomorphism problem of scattered and word-automatic linear orders.

References

- 1 V. Bárány, E. Grädel, and S. Rubin. Automata-based presentations of infinite structures. In *Finite and Algorithmic Model Theory*, pages 1–76. Cambridge University Press, 2011.
- 2 S.L. Bloom and Z. Ésik. The equational theory of regular words. *Information and Computation*, 197:55–89, 2005.
- 3 A. Blumensath and E. Grädel. Automatic Structures. In *LICS'00*, pages 51–62. IEEE Computer Society Press, 2000.
- 4 A. Blumensath and E. Grädel. Finite presentations of infinite structures: Automata and interpretations. *Theory of Computing Systems*, 37(6):641–674, 2004.
- 5 G. Braun and L. Strüngmann. Breaking up finite automata presentable torsion-free abelian groups. *International Journal of Algebra and Computation*, 21(8):1463–1472, 2011.

- 6 G. Cantor. Beiträge zur Begründung der transfiniten Mengenlehre, I. *Math. Annalen*, 46:481–512, 1895.
- 7 A. Carayol and Z. Ésik. The FC-rank of a context-free language. arXiv:1202.6275, February 2012.
- 8 B. Courcelle. Frontiers of infinite trees. *RAIRO - Theoretical Informatics*, 12(4):319–337, 1978.
- 9 M. Droste, W. Kuich, and H. Vogler, editors. *Handbook of Weighted Automata*. EATCS Monographs in Theoretical Computer Science. Springer, 2009.
- 10 M. Droste and D. Kuske. Weighted automata. To appear in the forthcoming handbook AutoMathA, 2012.
- 11 A. Durand-Gasselin and P. Habermehl. Ehrenfeucht-Fraïssé goes elementarily automatic for structures of bounded degree. In *STACS'12*, pages 242–253. Dagstuhl Publishing, 2012.
- 12 C.C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Am. Math. Soc.*, 98:21–51, 1961.
- 13 D.B.A. Epstein, J.W. Cannon, D.F. Holt, S.V.F. Levy, M.S. Paterson, and W.P. Thurston. *Word Processing In Groups*. Jones and Bartlett Publishers, Boston, 1992.
- 14 Z. Ésik. An undecidable property of context-free linear orders. *Inform. Processing Letters*, 111(3):107–109, 2011.
- 15 St. Heilbrunner. An algorithm for the solution of fixed-point equations for infinite words. *RAIRO – Theoretical Informatics*, 14(2):131–141, 1980.
- 16 B.R. Hodgson. On direct products of automaton decidable theories. *Theoretical Computer Science*, 19:331–335, 1982.
- 17 M. Huschenbett. The rank of tree-automatic linear orderings. arXiv:1204.3048, 2012.
- 18 S. Jain, B. Khoussainov, Ph. Schlicht, and F. Stephan. Tree-automatic scattered linear orders. Manuscript, 2012.
- 19 B. Khoussainov and A. Nerode. Automatic presentations of structures. In *Logic and Computational Complexity*, Lecture Notes in Comp. Science vol. 960, pages 367–392. Springer, 1995.
- 20 B. Khoussainov and A. Nerode. Open questions in the theory of automatic structures. *Bulletin of the EATCS*, 94:181–204, 2008.
- 21 B. Khoussainov, A. Nies, S. Rubin, and F. Stephan. Automatic structures: richness and limitations. *Log. Methods in Comput. Sci.*, 3(2), 2007.
- 22 B. Khoussainov, S. Rubin, and F. Stephan. Automatic linear orders and trees. *ACM Transactions on Computational Logic*, 6(4):675–700, 2005.
- 23 D. Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *International Journal of Algebra and Computation*, 4(3):405–425, 1994.
- 24 D. Kuske, J. Liu, and M. Lohrey. The isomorphism problem on classes of automatic structures with transitive relations. *Transactions of the AMS*, 2011. Accepted.
- 25 M. Lohrey and Ch. Mathissen. Isomorphism of regular trees and words. In *ICALP'11*, Lecture Notes in Comp. Science vol. 6756, pages 210–221. Springer, 2011.
- 26 Y. Matijasevich. *Hilbert's Tenth Problem*. Foundations of Computing Series. MIT Press, 1993.
- 27 A. Nies. Describing groups. *Bulletin of Symbolic Logic*, 13(3):305–339, 2007.
- 28 S. Rubin. *Automatic Structures*. PhD thesis, University of Auckland, 2004.
- 29 S. Rubin. Automata presenting structures: A survey of the finite string case. *Bulletin of Symbolic Logic*, 14:169–209, 2008.
- 30 W. Thomas. On frontiers of regular trees. *RAIRO – Theoretical Informatics*, 20(4):371–381, 1986.